Κλάση MinMaxPlayer

Η κλάση αυτή υλοποιεί τον παίκτη που παιζει με τον min-max αλγόριθμο. Ο αλγόριθμος αυτός υλοποιείται όπως ζητείται από την εκφώνηση. Πιο συγκεκριμένα:

1. Καλείται στην main η συνάρτηση getNextMove η οποία δημιουργεί τον κόμβο ρίζα του δέντρου. Καλεί την συνάρτηση δημιουργίας του δέντρου. Καλεί την chooseMinMaxMove η οποία επιστρέφει την καλύτερη κίνηση και τέλος καλεί την move η οποία εκτελεί την επιλεγμένη κίνηση.
2. Η συνάρτηση createMySubtree δέχεται ως όρισμα τη ρίζα του δέντρου και και δημιουργεί τους κόμβους βάθους ένα ανάλογα με τις δικές μας διαθέσιμες κινήσεις. Για την εύρεση των διαθέσιμων κινήσεων μας έχει δημιουργηθεί μία συνάρτηση possibleMoves. Σε ένα βρόχο για κάθε πιθανή κίνηση δημιουργείται ένα αντίγραφο του παίκτη μας έτσι ώστε να γίνει η κίνηση και καλείται η συνάρτηση createMySubtree για να δημιουργήσει τους κόμβους βάθους 2 που αντιστοιχούν σε κάθε κόμβο βάθους 1. Κάθε κόμβος βάθους 1 που δημιουργείται παίρνει τις κατάλληλες τιμές της κλάσης από τους setters και προστίθεται στο ArrayList children της ρίζας.
3. Η συνάρτηση createOpponentSubtree δέχεται ως όρισμα τους κόμβους βάθους 1 από την createMySubtree και δημιουργεί ανάλογα με τις διαθέσιμες κινήσεις της κάθε κατάστασης τους κόμβους βάθους 2 και τους προσθέτει στο ArrayList children του κάθε κόμβου βάθους 1 για τον οποίο καλείται η συνάρτηση. Έτσι, ολοκληρώνεται το δέντρο.
4. Εφόσον το δέντρο έχει πια ολοκληρωθεί καλειται η συνάρτηση chooseMinMaxMove οποία δέχεται ως όρισμα τον κόμβο ρίζα του δέντρου άρα έχει πρόσβαση σε όλους τους κόμβους του δέντρου. Για κάθε κόμβο βάθους 1 ,λοιπόν, συγκρίνει τις τιμές moveEvaluation τον παιδιών του, διαλέγει τη μικρότερη και την τοποθετεί ως moveEvaluation του κόμβου βάθους 1. Ύστερα συγκρίνει τις τιμές των μεταβλητών moveEvaluation των κόμβων βάθους 1, επιλέγει αυτόν με τη μεγαλύτερη και επιστρεφει τη ζαριά που αντιπροσωπεύει αυτός ο κόμβος.

Επιπλέον, για διευκόλυνση της των υπολογισμών της συνάρτησης evaluate, χρησιμοποιούνται και οι παρακάτω συναρτήσεις, η οποίες χρησιμοποιήθηκαν και στο προηγούμενο παραδοτέο για τον ίδιο λόγο.

- η playersDistance και η playersFromZero που επιστρέφουν την απόσταση των παικτών σε float από τον άλλο παίκτη και από το κέντρο των αξόνων αντίστοιχα. Η δεύτερη, συγκεκριμένα, χρησιμοποιείται προκειμένου να μην απομακρύνεται ο ευρετικός παίκτης ιδιαίτερα από το κέντρο του ταμπλό.

- η tilesDistance και η weaponDistance που επιστρέφουν την απόσταση (σε int) των παικτών από τον άλλο παίκτη ή κάποιο όπλο αντίστοιχα σε πλακίδια, δηλαδή σε πόσες κινήσεις θα έφτανε ο ευρετικός παίκτης τη συγκρινόμενη ποσότητα.

5. Όσον αφορά στη συνάρτηση evaluate, επιλέχθηκε η υλοποίηση 2 συναρτήσεων, μια για καθένα από τους δύο παίκτες, προς αποφυγή αύξησης της πολυπλοκότητας του κώδικα της evaluate.

    - evaluate (min - max player) : η οποία βασίζεται σχεδόν εξ' ολοκλήρου στην evaluate του heuristic player με μικρές αλλαγές. Μια εξ αυτών αποτελεί η αφαίρεση του πεδίου ορατότητας, καθώς πλέον ο παίκτης δύναται να σκανάρει ολόκληρο το ταμπλό. Επιπρόσθετα, επιλέχθηκε μέχρι ο παίκτης να φτάσει στην περίμετρο των παγίδων, να κινείται μόνο προς την κεντρική περιοχή. Η υπόλοιπη υλοποίηση δεν διαφέρει καθόλου σε ουσία από την υλοποίηση της evaluate του heuristic player. Σε κάθε περίπτωση όμως, ο κώδικας επεξηγείται με ακριβή σχόλια πριν από κάθε συνθήκη.
    - evaluate (opponent) : η οποία είναι πιστή μίμηση της evaluate του min - max player με την προφανή διαφορά πως όλοι οι έλεγχοι αυτή τη φορά γίνονται για τον αντίπαλο.

Τέλος, η συνάρτηση statistics χρησιμοποιείται όπως και στον Heuristic player, ενώ και η move χρησιμοποιείται αντίστοιχα ίδια με τη διαφορά ότι πλέον δεν υπάρχει ανάγκη για αποθήκευση όλων των δυνατών κινήσεων, με τις αντίστοιχες τιμές της evaluate, σε έναν χάρτη, όπως δηλαδή συνέβαινε στον Heuristic Player. Ακόμη, η kill ως static μπορεί και χρησιμοποιείται μέσω της κλάσης Heuristic Player.

Κλάση game

Η κλάση game έχει τροποποιηθεί έτσι ώστε να υποστηρίζει τον minMaxPlayer. Ο πρώτος παίκτης είναι τύπου minMaxPlayer ενώ ο δεύτερος τύπου Player. Και οι δύο ξεκινούν από την κάτω δεξιά θέση, έχει προστεθεί συνθήκη που τερματίζει το παιχνίδι εάν το σκορ κάποιου παίκτη γίνει αρνητικό. Τέλος για τον minMaxPlayer καλείται η συνάρτηση getNextMove μέσα στην οποία καλείται η move για να υλοποιηθεί η κίνηση του παίκτη. Τέλος,όπως και στο προηγούμενο παραδοτέο, έχουμε συμπεριλάβει σε σχόλια μία έκδοση της game για να τρέχουν πάνω από ένα παιχνίδια σε σχόλιο στο τέλος της main. Η μεταβλητή που ορίζει τον αριθμό των γύρων είναι η gameRounds και έχει οριστεί από εμάς σε 1000 αρχικά.
Προκειμένου να τρέξει ο κώδικας με τα πολλαπλά παιχνίδια, χρειάζεται να διαγραφεί ο κώδικας του ενός παιχνιδιού καθώς δεν μπορεί να γίνει ολόκληρος σχόλιο καθώς περιέχει και ο ίδιος σχόλια.

Κλάση Node

Η κλάση Node χρησιμοποιείται για να φτιαχτεί το δέντρο αντιπροσωπεύει τους κόμβους του. Κάθε κόμβος αντιπροσωπεύει μία κίνηση περιέχει τον αριθμό της στο ζάρι και την βαθμολογία της. Επίσης ορίζεται η δομή children για τους κόμβους βάθους ένα και μηδέν. Ακόμη, η kill ως static μπορεί και χρησιμοποιείται μέσω της κλάσης Heuristic Player.

> Διευκρίνιση : στην αναφορά έχουμε περιγράψει τη γενική ιδέα και την στρατηγική, περισσότερες λεπτομέρειες και η ροή του του κώδικα αναφέρονται λεπτομερώς στα σχόλια του κώδικα.